**TUGAS 4 PRAK DESAIN WEB A**

MEMBUAT APLIKASI MANAJEMEN DAFTAR TUGAS (To-Do List) INTERAKSI

**

Anavalis Ridho Abdee Nugroho (4523210012)

 Prak Desain Web A

1 - Oktober - 2024

Dosen Pembimbing :

Adi Wahyu Pribadi, S.Si., M.Kom.

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

* Pendahuluan
* Praktikum ini bertujuan untuk melatih kemampuan mahasiswa dalam membangun aplikasi berbasis web dengan menggunakan bahasa pemrograman JavaScript. Secara khusus, mahasiswa diharapkan mampu memahami dan menerapkan konsep-konsep dasar JavaScript seperti manipulasi DOM (Document Object Model), penggunaan class dan object, serta event handling untuk membuat aplikasi yang interaktif.
* Aplikasi yang akan dibuat dalam praktikum ini adalah **Aplikasi Manajemen Daftar Tugas (To-Do List)**. Aplikasi ini memungkinkan pengguna untuk menambahkan tugas baru, menandai tugas yang telah diselesaikan, serta menghapus tugas dari daftar. Selain itu, terdapat fitur validasi input untuk memastikan pengguna tidak dapat menambahkan tugas kosong, serta opsi untuk memfilter tugas berdasarkan statusnya (selesai atau belum selesai).
* Teori dasar yang diterapkan dalam praktikum ini meliputi:

1. JavaScript DOM Manipulation: Teknik untuk mengambil, memanipulasi, dan memperbarui elemen-elemen HTML secara dinamis.

2. Class dan Object: Penerapan konsep pemrograman berorientasi objek (OOP) dalam JavaScript untuk membangun struktur kode yang lebih modular dan terorganisir.

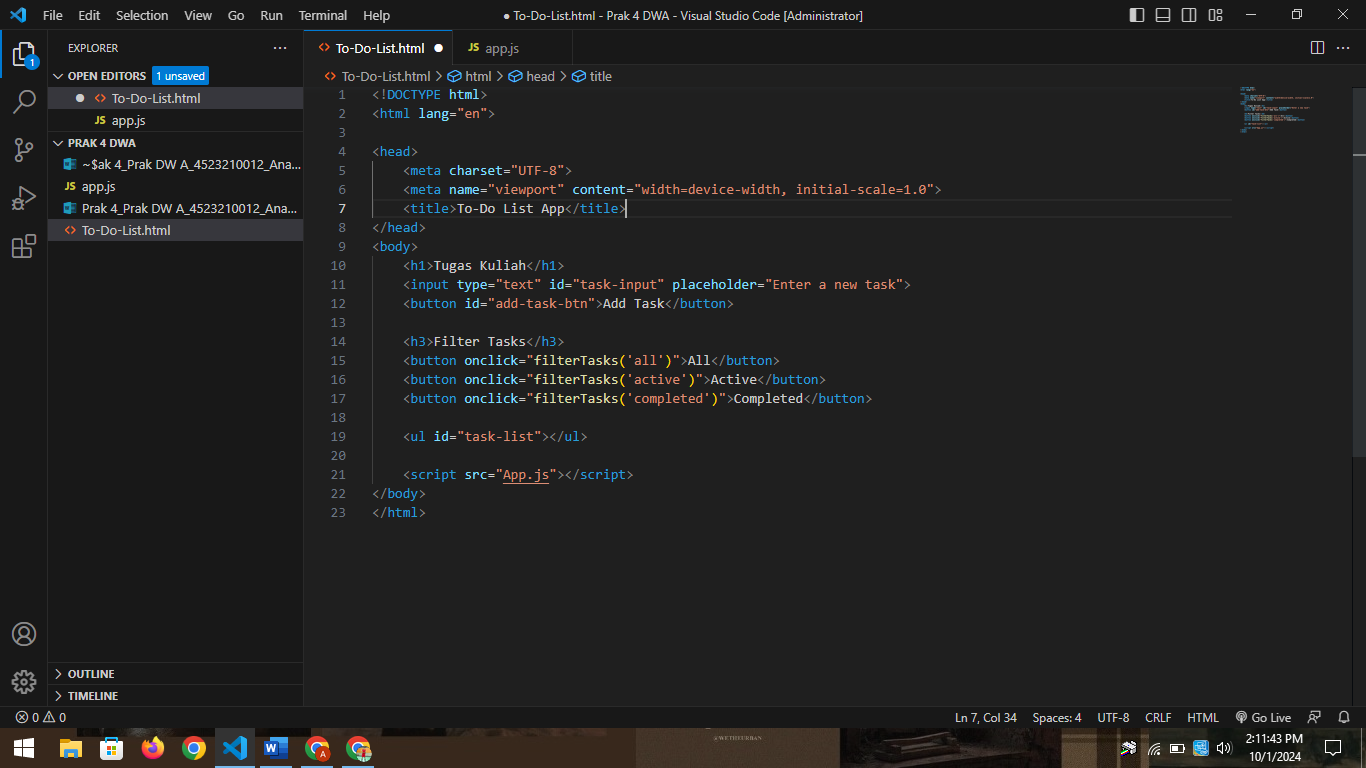
3. Event Handling: Mekanisme untuk menangani interaksi pengguna dengan antarmuka aplikasi, seperti klik tombol atau perubahan status checkbox, yang memungkinkan aplikasi menjadi lebih interaktif.

* Langkah Pengerjaan

Berikut adalah tahapan dalam mengerjakan praktikum pembuatan **Aplikasi To-Do List** yang interaktif menggunakan HTML, CSS, dan JavaScript:

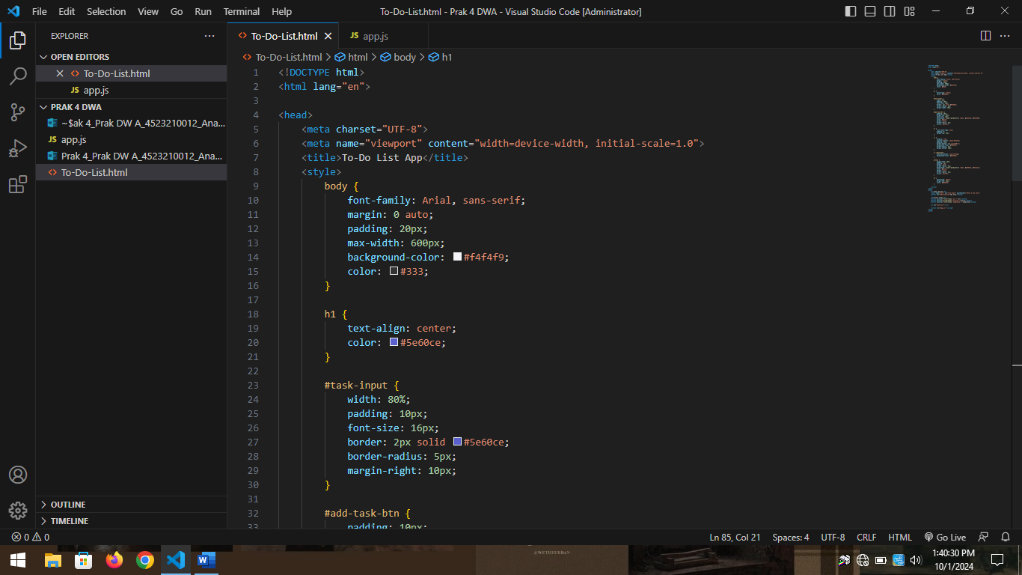
**1.** Menyiapkan Struktur HTML

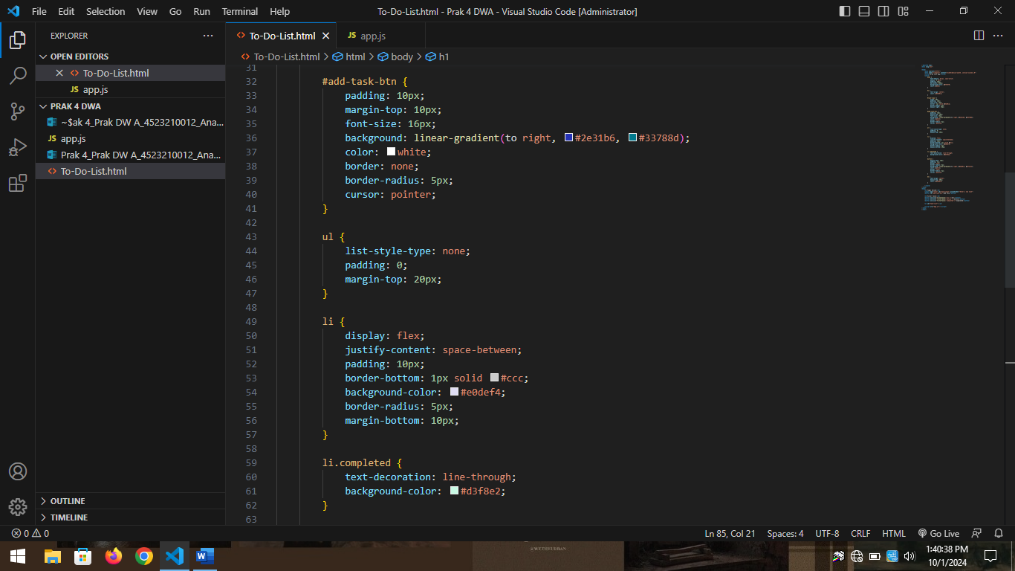
Langkah pertama adalah membuat file HTML yang berfungsi sebagai kerangka utama aplikasi. Pada bagian ini, saya membuat elemen-elemen dasar seperti input teks untuk menambahkan tugas, tombol untuk menambah tugas, serta daftar tugas yang akan ditampilkan.

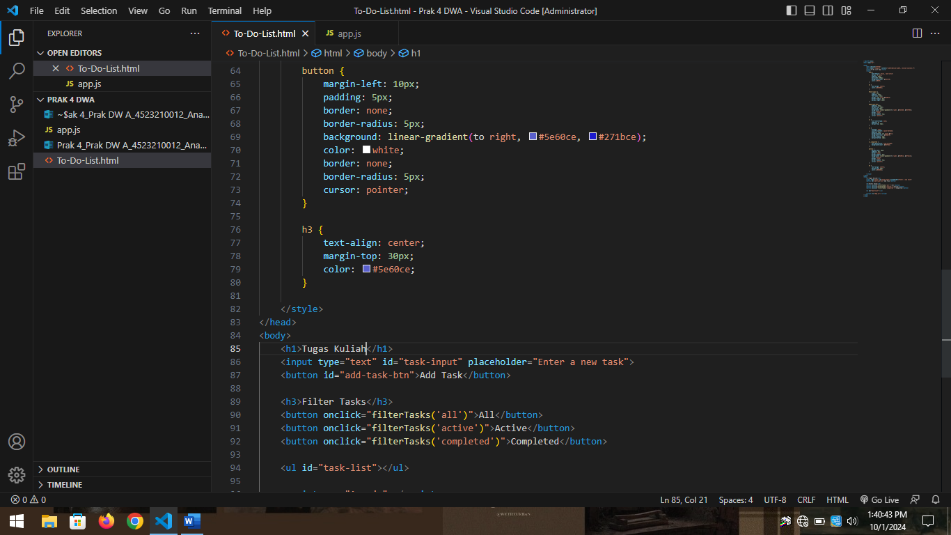


#### 2. Membuat Desain dengan CSS

Setelah struktur HTML selesai, langkah berikutnya adalah menambahkan gaya menggunakan CSS untuk membuat aplikasi lebih menarik dan mudah digunakan.



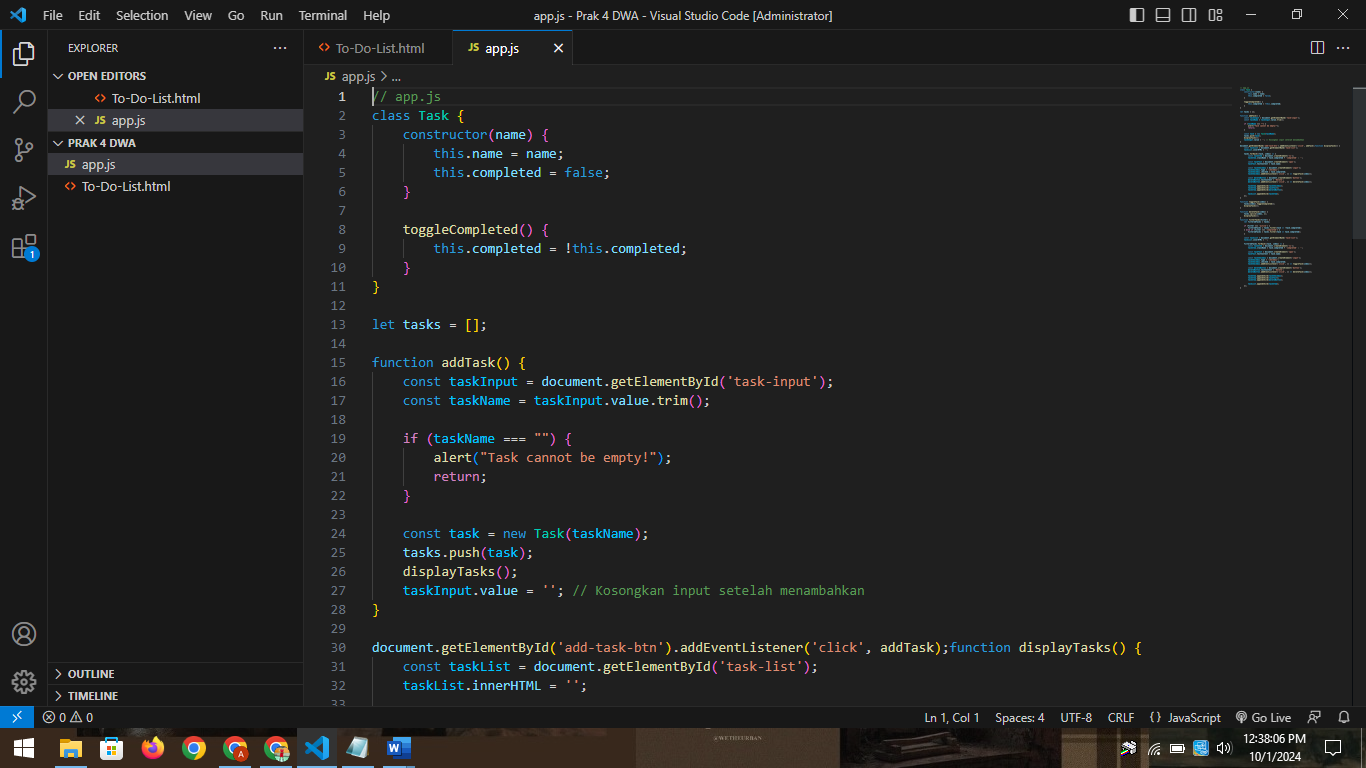


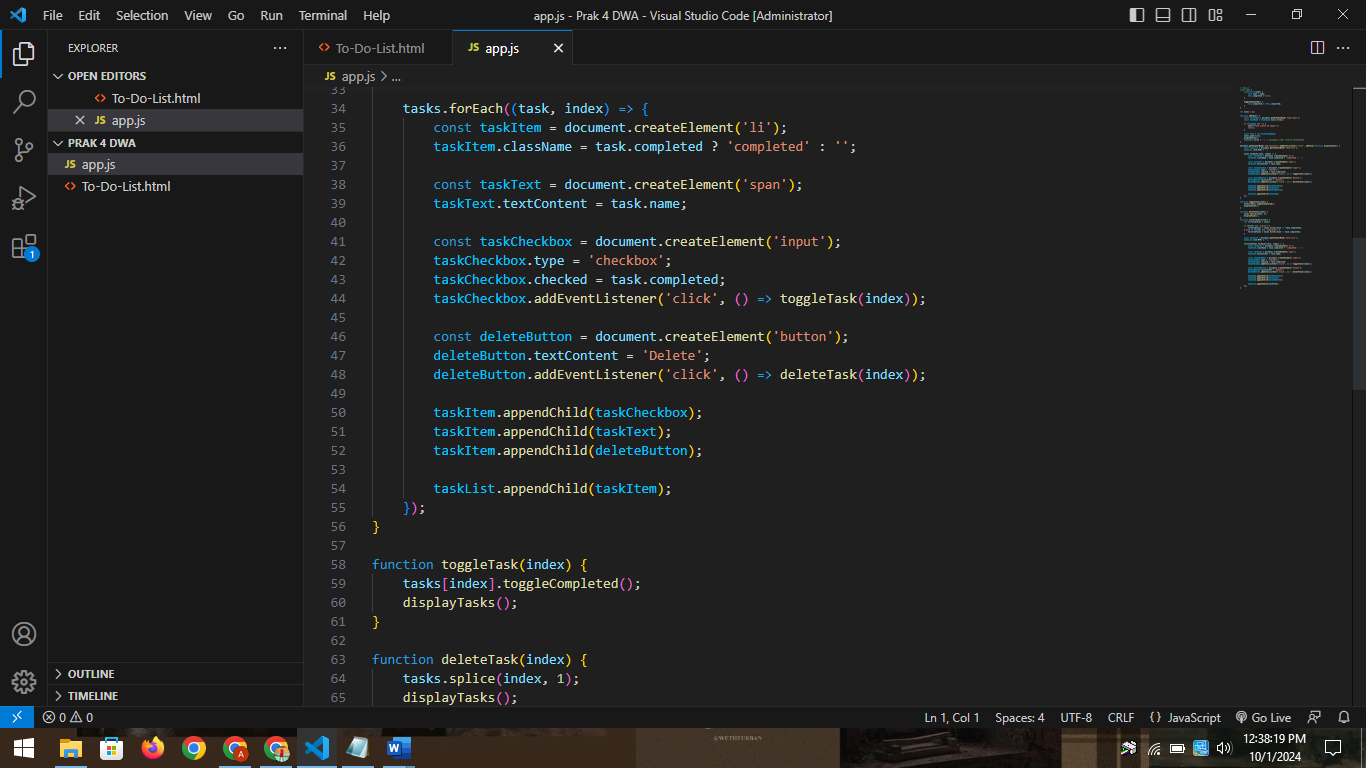


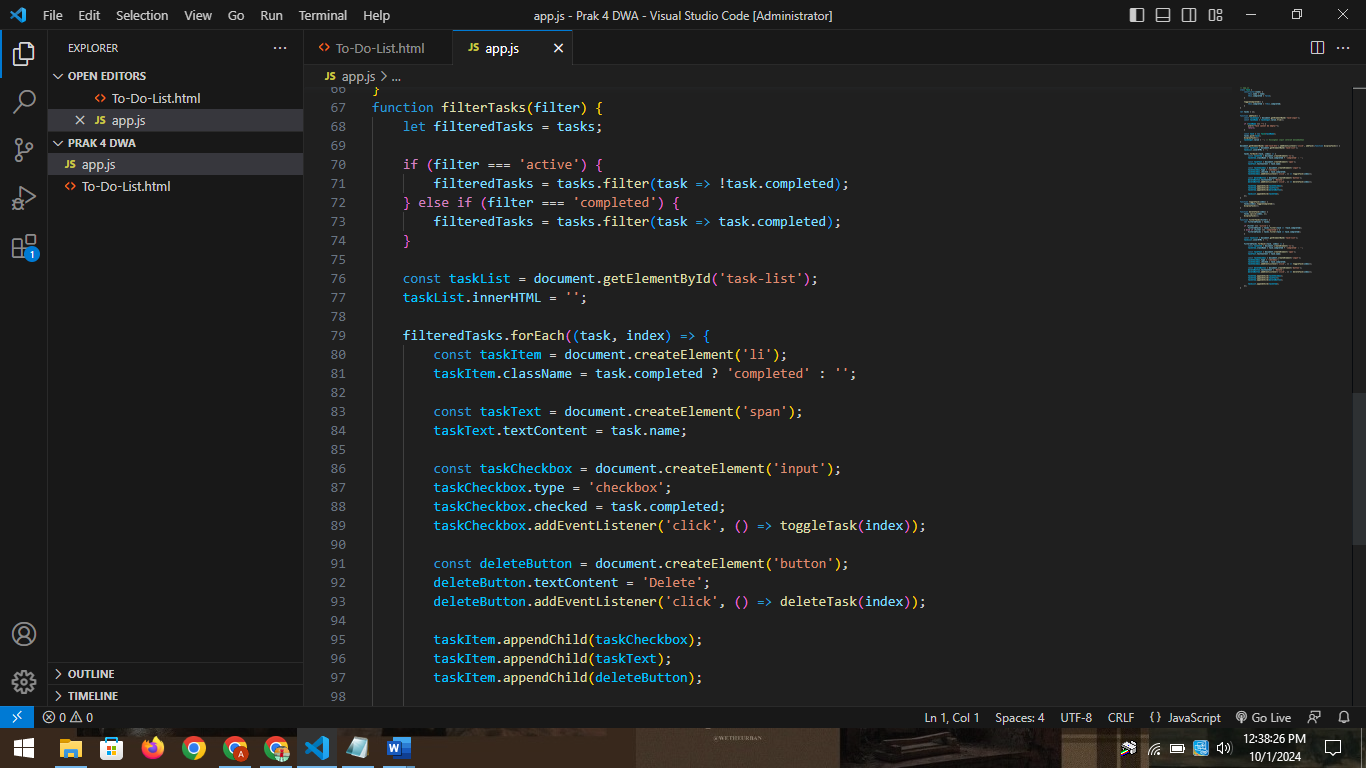
#### 3. Implementasi Fitur Menambahkan Tugas

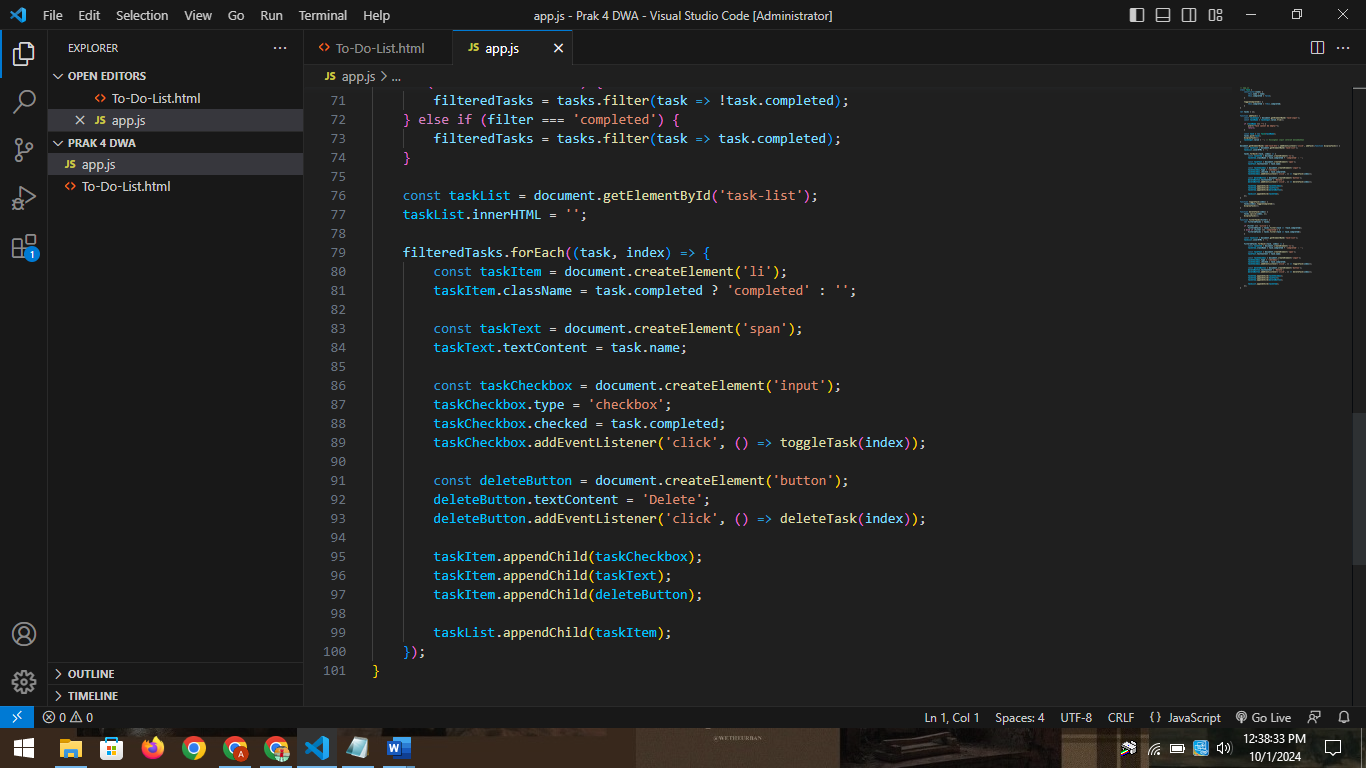
Setelah desain selesai, saya mulai mengimplementasikan fungsi JavaScript untuk menambahkan tugas ke daftar ketika pengguna menekan tombol “Add Task”. Pada tahap ini, saya juga menambahkan validasi agar pengguna tidak dapat menambahkan tugas kosong.

**Potongan kode JavaScript untuk menambahkan tugas:**









**Class Task dan Fungsinya:**

* Class Task adalah representasi dari tugas yang akan dikelola dalam aplikasi. Setiap instance dari Task memiliki dua properti:
  + name: Nama atau deskripsi tugas.
  + isCompleted: Status apakah tugas sudah selesai (default: false).
* Fungsi class ini untuk menyimpan informasi tentang tugas secara struktural.

**Fungsi addTask:**

* Fungsi ini digunakan untuk menambahkan tugas baru ke dalam array tasks yang menyimpan semua daftar tugas. Fungsi ini membuat objek Task baru dan menambahkannya ke array, kemudian menampilkan kembali seluruh daftar tugas yang ada.
* Validasi dilakukan untuk memastikan input tidak kosong sebelum tugas ditambahkan.

**Fungsi displayTasks:**

* Fungsi ini bertanggung jawab untuk memperbarui tampilan daftar tugas di DOM. Setiap tugas akan ditampilkan dalam bentuk elemen <li> dengan checkbox untuk menandai status selesai dan tombol untuk menghapus tugas.
* Manipulasi DOM digunakan di sini untuk membuat elemen-elemen HTML yang baru sesuai dengan isi array tasks.

**Fungsi toggleTask:**

* Fungsi ini digunakan untuk mengubah status selesai atau belum selesai pada sebuah tugas. Status isCompleted diubah berdasarkan interaksi pengguna dengan checkbox.

**Fungsi deleteTask:**

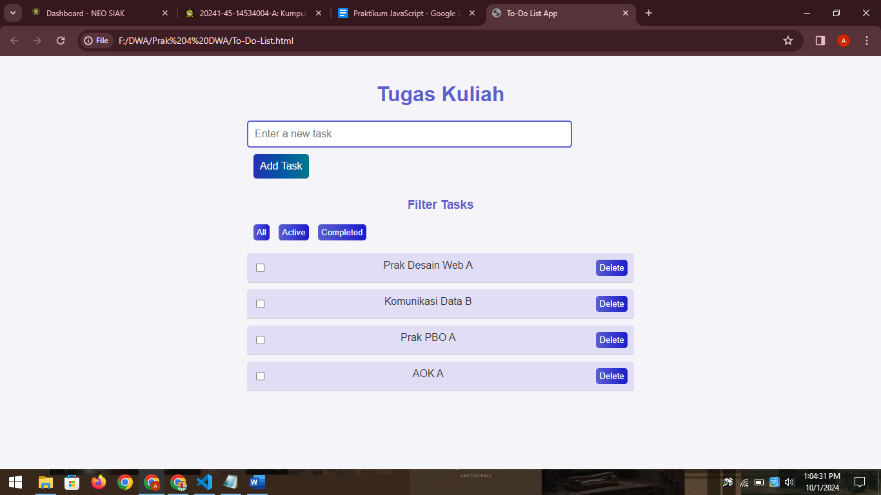
* Fungsi ini menghapus tugas dari array tasks dan kemudian memperbarui tampilan DOM setelah tugas dihapus.

**Fungsi filterTasks:**

* Fungsi ini memungkinkan pengguna untuk memfilter tugas yang ditampilkan berdasarkan statusnya (semua tugas, tugas selesai, atau tugas belum selesai).

### Manipulasi DOM dan Event Handling

* **Manipulasi DOM:**
  + Digunakan di fungsi displayTasks untuk membuat elemen-elemen HTML dinamis seperti <li>, checkbox, dan tombol "Delete". Hal ini memungkinkan pembaruan real-time pada tampilan daftar tugas berdasarkan interaksi pengguna.
* **Event Handling:**
  + Pada tombol **“Add Task”**, event handling dilakukan menggunakan addEventListener('click', addTask) yang memicu fungsi addTask ketika tombol diklik.
  + Pada checkbox, event handling menangani perubahan status tugas (checked/unchecked). Fungsi toggleTask dipanggil saat checkbox diubah, menggunakan data index dari tugas.
  + Pada tombol **“Delete”**, event handling digunakan untuk menghapus tugas dari daftar, dengan memanfaatkan atribut data-index untuk mengetahui tugas yang harus dihapus.
* Hasil Uji Coba



* Kesimpulan

Melalui praktikum ini, saya telah mendapatkan pemahaman yang lebih mendalam mengenai HTML, JavaScript, dan CSS. Implementasi langsung dalam pembuatan aplikasi To-Do List memberikan gambaran praktis mengenai bagaimana teknologi ini dapat diterapkan dalam dunia kerja nantinya. Semoga pengetahuan yang diperoleh dari praktikum ini dapat terus bermanfaat di masa mendatang, terutama dalam pengembangan aplikasi berbasis web.